(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1894894 B
(45)授权公告日 2011.07.27

(21)申请号 200480031653.0

(22)申请日 2004.11.05

(30)优先权数据
60/517,934 2003.11.05 US
10/981,900 2004.11.04 US

(85)PCT申请进入国家阶段日
2006.04.26

(86)PCT申请的申请数据
PCT/US2004/036782 2004.11.05

(87)PCT申请的公布数据
WO2005/048508 EN 2005.05.26

(73)专利权人 瞻博网络公司
地址 美国加利福尼亚州

(72)发明人 B·辛格 A·P·辛格 V·帕克森

(74)专利代理机构 北京戈程知识产权代理有限公司 11314
代理人 程伟

(51)Int.Cl.
H04L 12/26(2006.01)

(56)对比文件
US 6252851 B1,2001.06.26, 全文.
US 5163046 A,1992.11.10, 全文.
US 6560243 B1,2003.05.06, 全文.

审查员 曹娟

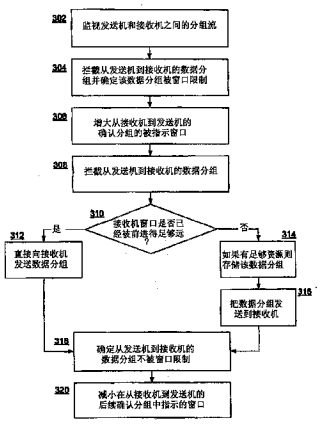
权利要求书 4 页 说明书 9 页 附图 7 页

(54)发明名称

传输控制协议流控制的透明优化方法及系统

(57)摘要

一种不用侵入 TCP 的核心算法而优化传输控制协议 (TCP) 流控制的系统和方法。相对靠近发送机局域网 (LAN) 的控制模块自动识别已经变成窗口限制的分组流。在该分组流已被识别为窗口控制的之后，相对靠近该发送机局域网的控制模块和相对靠近接收机局域网的另一个控制模块通过增大接收机的确认分组中指出的窗口大小来优化该分组流。这两个控制模块同步操作，以便透明地管理该发送机和该接收机之间的该分组流。



CN 1894894 B

1. 一种优化发送机和接收机之间的传输控制协议 TCP 流控制的方法，其包括：

用位于所述发送机和所述接收机之间的第一控制模块，拦截从所述发送机到所述接收机的数据分组，其中所述第一控制模块与所述发送机相关联；

用所述第一控制模块，基于所述接收机窗口的当前大小，确定所述数据分组被窗口限制；

用所述第一控制模块，响应于确定所述数据分组被窗口限制，把所述数据分组表征为被窗口限制；

用所述第一控制模块，发送被表征的数据分组至所述接收机；

用位于所述发送机和所述接收机之间的与所述接收机相关联的第二控制模块，拦截所述被表征的数据分组，并且确定指示数据分组被窗口限制的所述数据分组的表征；

用所述第二控制模块除去所述数据分组的表征，并且将数据分组发送至所述接收机；

用所述第二控制模块拦截来自于所述接收机的确认分组；

用所述第二控制模块，响应于确定所述数据分组被窗口限制的表征，增大所接收的确认分组中指示的窗口；和

用所述第二控制模块，从所述第二控制模块向所述发送机发送具有所述被增大的窗口指示的所述确认分组。

2. 根据权利要求 1 所述的方法，还包括利用所述第一控制模块执行步骤：

响应于向所述发送机发送所述确认分组，拦截来自于所述发送机的另一个数据分组；

确定所述接收机的窗口是否已经被前进，以便容纳来自于所述发送机的所述另一个数据分组；

如果所述接收机的窗口已经被前进、以便容纳所述另一个数据分组，则向所述接收机发送所述另一个数据分组；和

如果所述接收机的窗口没有被前进、以便容纳所述另一个数据分组，则使用与所述控制模块之一相关联的存储区存储所述另一个数据分组。

3. 根据权利要求 2 所述的方法，还包括：当来自于所述接收机的后续确认分组指示所述接收机的窗口已经被前进、以便容纳所述被存储的数据分组时，向所述接收机发送所述被存储的数据分组。

4. 根据权利要求 3 所述的方法，还包括：在向所述接收机发送所述被存储的数据分组之前，调整包括在所述被存储的数据分组中的时间戳。

5. 根据权利要求 1 所述的方法，还包括执行步骤：

用所述第一控制模块，拦截来自于所述发送机的另一个数据分组；

用所述第一控制模块，确定所述另一个数据分组不被窗口限制；

用所述第二控制模块，拦截来自于所述接收机的另一个确认分组；

用所述第二控制模块，响应于确定所述另一个数据分组不被窗口限制，减小在所述另一个确认分组中指示的窗口；和

从所述第二控制模块向所述发送机发送具有所述被减小的窗口指示的所述另一个确认分组。

6. 根据权利要求 5 所述的方法，还包括：

把所述另一个数据分组表征为不被窗口限制；以及

并向所述接收机发送所述被表征的数据分组。

7. 根据权利要求 5 所述的方法，其中，减小在所述另一个确认分组中指示的窗口的步骤包括：把在所述另一个确认分组中指示的窗口减小一个量，使得所述被指示窗口的顶端保持不变。

8. 根据权利要求 1 所述的方法，还包括：存储由所述接收机发送的一个或多个确认分组中指示的一个或多个窗口；并基于所存储一个或多个窗口来确定数据传输速率。

9. 根据权利要求 1 所述的方法，其中，确定所述数据分组被窗口限制的步骤包括：确定所述数据分组的序列号基本上接近于所述被指示窗口的顶端。

10. 根据权利要求 1 所述的方法，

其中，拦截来自于所述发送机的所述数据分组的步骤包括：在相对接近所述发送机的位置处拦截来自于所述发送机的所述数据分组；和

其中，拦截来自于所述接收机的所述确认分组步骤包括：在相对接近所述接收机的位置处拦截来自于所述接收机的所述确认分组。

11. 根据权利要求 1 所述的方法，还包括：

识别分组的大小，用于所述 TCP 流控制的优化；和

基于所识别的所述分组的大小，利用所述第一和第二控制模块中的一个确定分组大小调整，用于所述分组在所述接收机窗口的原始大小内、以与所述网络优化一致的传输率而到达所述接收机。

12. 根据权利要求 11 所述的方法，还包括：

确定所述分组中最高位的序列号和所述分组的当前的确认号之差；

通过将所述接收机窗口的原始大小除以所述被确定的差，而确定数值；和

将所述被确定的数值与所述被识别的分组大小相乘以确定分组大小调整。

13. 一种用于优化发送机和接收机之间的传输控制协议 TCP 流控制的系统，该系统包括位于所述发送机和接收机之间的第一和第二控制模块，其特征在于，

所述第一控制模块与所述发送机相关联并被配置为：

拦截来自于所述发送机的数据分组；

确定所述数据分组被窗口限制；

响应于确定所述数据分组被窗口限制，把所述数据分组表征为被窗口限制；以及

发送被表征的数据分组至所述接收机；以及

所述第二控制模块与所述接收机相关联并被配置为：

拦截所述被表征的数据分组，并且确定指示数据分组被窗口限制的所述数据分组的表征；

除去所述数据分组的表征，并且将数据分组发送至所述接收机；

拦截来自于所述接收机的确认分组；

响应于确定所述数据分组被窗口限制，增大在所接收的确认分组中指示的窗口；以及

从所述第二控制模块向所述发送机发送具有所述被增大的窗口指示的所述确认分组。

14. 根据权利要求 13 所述的系统，其中，所述第一控制模块：

响应于向所述发送机发送所述确认分组，拦截来自于所述发送机的另一个数据分组；

确定所述接收机的窗口是否已经被前进、以便容纳来自于所述发送机的所述另一个数

据分组；

如果所述接收机的窗口已经被前进、以便容纳所述另一个数据分组，则向所述接收机发送所述另一个数据分组；以及

如果所述接收机的窗口还没有被前进、以便容纳所述另一个数据分组，则把所述另一个数据分组存储在与所述控制模块相关联的存储区中。

15. 根据权利要求 14 所述的系统，其中，当来自于所述接收机的后续确认分组指示所述接收机的窗口已经被前进、以便容纳所述被存储的数据分组时，所述第一控制模块把所述被存储的数据分组发送到所述接收机。

16. 根据权利要求 15 所述的系统，其中，所述第一控制模块在向所述接收机发送所述被存储的数据分组之前，调整包括在所述被存储的数据分组中的时间戳。

17. 根据权利要求 13 所述的系统，其中，所述第一控制模块：

接收来自于所述发送机的另一个数据分组；

确定所述另一个数据分组没有被窗口限制；以及

其中所述第二控制模块：

接收来自于所述接收机的另一个确认分组；

响应于确定所述另一个数据分组不被窗口限制，减小在所述另一个确认分组中指示的窗口；

并且向所述发送机发送具有所述被减小的窗口指示的所述另一个确认分组。

18. 根据权利要求 17 所述的系统，其中，所述第一控制模块：

把所述另一个数据分组表征为不被窗口限制；以及

向所述接收机发送所述被表征的数据分组。

19. 根据权利要求 17 所述的系统，其中，所述第一控制模块把在所述另一个确认分组中指示的窗口减小一个量，使得所述被指示窗口的顶端保持不变。

20. 根据权利要求 13 所述的系统，其中，所述第二控制模块：

把由所述接收机发送的一个或多个确认分组中指示的一个或多个窗口存储在存储区中；以及

基于所存储一个或多个窗口来确定数据传输速率。

21. 根据权利要求 13 所述的系统，其中，通过确定所述数据分组的序列号基本上接近于所述被指示窗口的顶端，所述第二控制模块确定所述数据分组被窗口限制。

22. 根据权利要求 13 所述的系统，其中，所述第一或第二控制模块识别分组的大小，用于所述 TCP 流控制的优化，并基于所识别的经过所述网络优化的所述分组的大小，确定分组大小调整，用于所述分组在所述接收机窗口的原始大小内、以与所述网络优化一致的传输率而到达所述接收机。

23. 根据权利要求 22 所述的系统，其中，所述第一或第二控制模块通过确定所述分组中最高位的序列号和所述分组的当前的确认号之差；通过将所述接收机窗口的原始大小除以所述被确定的差，而确定数值；和将所述被确定的数值与所述被识别的分组大小相乘，来确定所述分组大小调整。

24. 一种使用位于发送机和接收机之间的一个控制模块来优化传输控制协议 TCP 流控制的方法，包括：

拦截从所述发送机到所述接收机的数据分组；

基于所述接收机窗口的当前大小，确定所述数据分组被窗口限制；

拦截来自于所述接收机的确认分组；

响应于确定所述数据分组被窗口限制，增大接收的所述确认分组中指示的窗口；以及

向所述发送机发送具有被增大的窗口指示的确认分组。

25. 一种用于优化发送机和接收机之间的传输控制协议 TCP 流控制的系统，所述系统包括位于所述发送机和接收机之间的一个控制模块，其特征在于，所述控制模块：

拦截来自所述发送机的数据分组；所述控制模块确定所述数据分组被窗口限制；

拦截来自于所述接收机的确认分组；

响应于确定所述数据分组被窗口限制，增大接收的所述确认分组中指示的窗口；以及

向所述发送机发送具有被增大的窗口指示的确认分组。

传输控制协议流控制的透明优化方法及系统

[0001] 相关申请的交叉参考：

[0002] 本申请要求申请日为2003年11月5日的美国临时申请号60/517,934的优先权，其内容被并入在此引作参考。

[0003] 本申请涉及申请日为2004年11月4日的美国专利申请号_/__,__，标题为" Transparent Optimization for Transmission ControlProtocol Initial Session Establishment"，其内容被并入在此引作参考。

技术领域

[0004] 本发明涉及通过网络进行联网和数据通信的领域，并且尤其涉及对于传输控制协议(TCP)流控制的透明优化。

背景技术

[0005] 传输控制协议(TCP)是在通信网络上被最广泛应用和可靠的数据传输协议之一。传输控制协议的主要差别及其被被广泛应用的理由之一是用于共享当前TCP会话之间的带宽的鲁棒算法。TCP内的这种共享算法通常被称为"拥塞控制"，因为它试图通过自动回缩(scalingback)数据传输以匹配可用的带宽能力来避免网络拥塞的问题。如果每个数据传输会话都试图完全利用链路容量，则通过共享网络链路的多个并行和可靠的数据传输可能导致高度拥塞。这样的高度拥塞可能导致高度的分组丢失，其接着可能导致大量的分组重发，最终致使网络崩溃。TCP的拥塞控制算法通过自动确定有多少带宽可用并且通过与其它并行的TCP会话同等地共享总可用带宽来避免这个问题。因此，TCP的动态共享算法是通过分组交换互联网协议(IP)网络的数据通信的基础结构块(fundamental building block)，并且已经使得TCP/IP被用作通用的通信标准。

[0006] TCP用各种各样的内部算法来提供其拥塞控制的能力。这些算法包括流控制、慢起动、分组重排(reordering)、分组丢失检测、重发计 时器、以及大量的其它机理，用于基于网络情况动态地降低或提高数据传输速率。

[0007] 网络延迟(latency)是影响网络以及应用性能的一个普通问题。网络延迟可归因于几个因素，包括物理距离、;重复接收(hops)数量、交换机和路由器中继以及网络拥塞。因为这些因素不是一成不变的，所以网络可能在一段时间内具有不可预知的延迟。网络延迟的变化取决于链接使用的网络链路和传输介质所跨越的距离。例如，地铁区域内两个建筑物之间的局部高速专用线路可能经历5毫秒(ms)的单向延迟，而美国和欧洲之间的球形长途异步传输模式(ATM)链路可能具有从50ms到250ms之间任何数值的单向延迟。类似地，由于向轨道卫星发射信号以及返回的时间，卫星链路一般经历了大约240到300ms的单向延迟。

[0008] 关于网络应用的延迟冲击可以被直接回溯到TCP在网络延迟条件下的低效率。大多数网络应用能够被区分为基于"频繁访问(chatty)"的短交易应用或成批数据传输应用。成批数据传输应用一般发射数百千字节或兆字节的数据通过网络，其总传输时间在数秒、

数分钟、多数情况下是数小时中来测量。这类应用的例子包括网络文件系统、存档和存储应用、文件传输协议 (FTP) 传输、共享以及分发大工程或设计文档等。在这些应用中，普遍的性能瓶颈往往是网络上的延迟，其导致经由 TCP 的应用吞吐量降低。特别是，TCP 内的流控制算法往往导致较低的应用吞吐量和较高的应用响应时间。

[0009] TCP 的流控制算法是防止接收机接收大于其处理或缓存能力的数据的一种机理。例如，如果接收 TCP 栈具有能存储 16 千字节数据的缓存，则在任何时候都不允许发送机向接收机发射超过 16 千字节的数据。接收机在数据传输过程中连续地向发送机发送回确认，其表明接收机还能够接受多少附加数据。接收机能够接受的这些附加数据被称为″窗口指示″（或″窗口广告″），并且被包括为 TCP 报头中的字段。

[0010] 流控制算法有效率地操作，并且当接收机和发送机被短程低延迟链路分隔时不引入任何不必要的延迟。但是，随着接收机和发送机之间的距离和延迟被增加，发送数据分组和接着从接收机接收确认之间的往返时间 (RTT) 也增加了。因为流控制算法防止发送机在接收机还 未表明它准备好接收这些附加数据的时候向接收机发射数据，所以两个端点 (endpoint) 之间的 RTT 可能导致发送机延迟发送附加的数据分组，以等待来自于接收机的下一个确认。例如，如果接收机一次能够接受 16 千字节的数据，则发送机可以在数毫秒中发射全部的 16 千字节，然后花费等待确认的几个附加毫秒来开始发射下一个 16 千字节块。发送机正在等待确认的这个时段取决于带宽和链路延迟两者。基于延迟的 TCP 空闲时间可能导致单个 TCP 流与网络链路上实际可用的情况相比较而言，获得较低的吞吐量。因为 TCP 流不能利用全部的现有网络带宽，所以这些未利用的容量转化为较高的传输时间。

[0011] 流管线化被用于其数据传输速率受到限制的 TCP 连接，因为接收机上配置的最大 TCP 窗口的尺寸小于它传输数据所通过网络的带宽时延积，所以其能够传输数据。对于这个问题的传统解决方案是接收机通告或表明大于网络的带宽时延积的窗口大小。然而，由于实际的原因，这个所指出的窗口大小可能受到接收机和发送机上可用存储器的限制。从而，由操作系统开发商选择一个合理的值，并将其设置为窗口大小的缺省值。这个值适于局域网上的大多数 TCP 连接。但是当 TCP 连接建立在高延迟网络上的时候，这个值可能不够大。参与数据传输的计算机无法动态地发现并固定窗口尺寸小的问题。这是因为这个问题只能在发送机侧上的网络最长延迟段之前被可靠地处理数据的装置（其已知带宽和长延迟段当前的利用率）注意到。然而，在这个装置确定特殊的传输收到窗口限制之后，TCP 不向该装置提供方法，以通知接收机希望增大窗口的愿望。

[0012] 一种让这个装置任意增加窗口大小的解决方案产生了不够好结果，因为它产生了大量窗口之外的数据。因此，当存在诸如单个分组丢失之类的小的网络错误时，这个解决方案可能导致所接收窗口之外的大量数据。接收窗口之外的数据可能在它到达接收机的时候被接收机丢弃。除了让发送机重发被丢弃数据之外，它还导致发送机把丢失误读为网络拥塞事件，并从而降低了其传输。

[0013] 所需要的是一种无需侵入 TCP 的核心算法而优化 TCP 的流控制的系统和方法，用于改进 TCP 会话的性能。

发明内容

[0014] 本发明是一种不侵入 TCP 的核心算法而优化 TCP 的流控制的系统和方法。本发明

双向地监视网络管道两侧的 TCP 流的状态。相对靠近发送机局域网 (LAN) 的控制模块基于 TCP 报头信息自动识别已经变成延迟限制（或″窗口限制″）的分组流。在分组流由于网络延迟已经被识别为窗口控制的之后，相对靠近发送机局域网的控制模块和相对靠近接收机局域网的另一个控制模块通过增加在接收机的确认分组中指出的窗口大小来优化该分组流。两个控制模块同步操作，以便透明地管理发送机和接收机之间的分组流。这个透明优化过程允许发送机最大化传输中的数据量，由此基本上最小化发送机等待来自接收机的附加确认的空闲时间。

[0015] 说明书中描述的特征和优点不是总括的，特别是，就附图、说明书和权利要求而言，许多附加的特征和优点对于本领域普通技术人员来说是显而易见的。而且应当注意，说明书中的使用的语言主要被选择用于易读和指导的目的，并且可能没有被选择用于叙述或限制本发明的主题。

附图说明

[0016] 图 1 是本发明能在其中运行的一个网络环境示例的图示。

[0017] 图 2A 和图 2B 是 TCP 的流控制算法示例的图示。

[0018] 图 3 是通过本发明的一个实施例实施的、用于优化 TCP 的流控制的方法的流程图。

[0019] 图 4 是由本发明的一个实施例实施、并且由相对靠近发送机的控制模块执行、以优化 TCP 的流控制的方法的流程图。

[0020] 图 5 是由本发明的一个实施例实施、并且由相对靠近接收机的控制模块执行、以优化 TCP 的流控制的方法的流程图。

[0021] 图 6 是说明没有通过本发明实施例优化的窗口限制的 TCP 交易示例的时序图。

[0022] 图 7 是说明通过本发明实施例优化的窗口限制的 TCP 交易示例的时序图。

具体实施方式

[0023] 现在将参考附图对本发明的优选实施例进行描述，其中，类似的附图标记指示相同或功能类似的部件。此外，附图中每个附图标记最左边的数字对应于首先使用该附图标记的附图。

[0024] 说明书中参考的“一个实施例”或“实施例”意味着特殊的特征、结构或特性被包括在本发明的至少一个实施例中。说明书中不同处所出现的短语“在一个实施例中”不一定全都指的是相同的实施例。

[0025] 随后的详细说明的某些部分按照计算机存储器中按数据位操作的算法和符号表示给出。这些算法说明和表示被那些数据处理领域内的技术人员用于更有效地向本领域的其它技术人员传达其工作实质。在此，算法通常被设想为产生期望结果的自相容步骤（指令）序列。这些步骤是需要物理量的物理操作。尽管不一定，但这些量通常采用能够被存储、传输、结合、比较及其它操作的电、磁或光信号的形式。有时候，主要是由于公共用途的原因，把这些信号称为比特、值、元素、符号、特性、术语、数量等等是方便的。而且不失一般性地，有时把需要物理量的物理操作的步骤的某些布置称为模块或编码装置也是方便的。

[0026] 然而应该考虑到，这些项以及类似项将与适当的物理量相关联、并且仅仅是被应用于这些量的方便的标签。除非特别指出，否则从以下论述可以明显看出，应当理解，在说

明书和论述中使用的诸如″处理″或″计算″或″确定″或″显示″或″确定″之类的术语指的是计算机系统或类似的电子计算装置的动作和处理，这类计算机系统或电子计算装置操作并转换在计算机系统存储器或寄存器或其它这类信息存储、传输或显示装置内被表示为物理（电子）量的数据。

[0027] 本发明的某些方面包括在此以算法形式描述的处理步骤和指令。应当注意，本发明的处理步骤和指令能够被具体实现为软件、固件或硬件，而且在其被具体实现为软件时，可以被下载保存，并可以从各种操作系统使用的不同平台来操作。

[0028] 本发明还涉及一种用于执行这里所描述的操作的装置。这个装置可以被特别地构造成用于所需要的目的，或者它可能包括由存储在计算机中的计算机程序来有选择地激活或重新配置的通用计算机。这类 计算机程序可以被存储在计算机可读的存储介质中，例如、但不限于任何类型的磁盘，包括软盘、光盘、CD-ROM、磁光盘、只读存储器 (ROMs)、随机存取存储器 (RAM)、EPROM、EEPROM、磁或光卡片、专用集成电路 (ASIC)、或者适于存储电子指令的任何媒介类型，并且每一个都与计算机系统总线相连。而且，说明书中涉及的计算机可以包括单个处理器，或者可以是采用多个处理器设计的结构，以便增加计算能力。

[0029] 这里示出的算法和显示不是固有地与任何特殊的计算机或其它设备相关。各种各样的通用系统也可以用于根据此处所给出的程序，或者它可以被证明为方便构造更专用的设备，以执行所要求的方法步骤。被要求用于这些各式各样的系统的结构将由下面的说明而变得明显。另外，本发明没有参考任何特殊的程序语言来描述。应当理解，各种各样的编程语言可以被用来实现在此所描述的本发明的目的，并且下面对专用语言的任何参考都被提供用于公开本发明的实现和最佳模式。

[0030] 另外，说明书中的使用的语言主要被选择用于易读和指导的目的，并且可不被选择用于叙述或限制本发明的主题。因此，本发明的公开内容意在说明而不是限制在随后的权利要求中所阐明的本发明的范围。

[0031] 图 1 是本发明能在其中运行的一个网络环境示例的图示。接收机 102 可以是用 TCP 向一个或多个端点发送数据或从一个或多个端点接收数据的任何装置。接收机 102 经由诸如 LAN 106 之类的通信网络被连接到与控制模块 104（例如，在控制装置中）相连。可选地，控制模块 104 没有经由 LAN 106 而直接与接收机 102 相连，或者在接收机 102 内被实现为程序模块。控制模块 104 经由诸如广域网 (WAN) 108 之类的另一个通信网络与一个或多个其它控制模块相连。即使图 1 示出了控制模块 104 被连接到一个其它的控制模块（即，控制模块 110），控制模块 104 也可能被连接到超过一个的控制模块。每个其它的控制模块还经由另一个通信网络被连接到发送机。例如，图 1 示出（例如在控制装置中的）了控制模块 110 经由 LAN 114 被连接到发送机 112。发送机 112 可以是用 TCP 向一个或多个端点发送数据或者从一个或多个端点接收数据的任何装置。而且，控制模块 110 能够不经由 LAN 114 而直接连接到发送机 112，或者可能在发送机 112 内被实现为程序模块。

[0032] 在本发明的一个实施例中，控制模块 104 与接收机 102 的相对位置比与发送机 112 的相对位置靠近，而控制模块 110 与发送机 112 的相对位置比与接收机 102 的相对位置靠近。因此，控制模块 104 物理上和逻辑上与接收机 102 相关联，而控制模块 110 物理上和逻辑上与发送机 112 相关联。

[0033] 已经在接收机 102 和发送机 112 之间建立 TCP 会话之后，TCP 开始提供流控制服

务，以防止发送机 112 溢出接收机 102 的缓存。TCP 流控制使发送机 112 发送应用层数据的速率与接收机 102 读取该数据的速率相匹配。TCP 通过使发送机 112 保持被称为接收窗口的变量来提供流控制。接收窗口被用来给予发送机 112 在接收机 102 处有多少空闲的缓冲区的显式信息。接收窗口是动态的；即，它在 TCP 会话的存在期间会变化。通过在它发送到发送机 112 的 TCP 分组的窗口中通告其接收窗口的当前值，接收机 102 告知发送机 112 它在连接缓存中有多少备用空间。通过将未确认数据的量保持成小于接收窗口的值，发送机 112 能够防止自身在接收机 102 处溢出接收机缓存。

[0034] 图 2A 和 2B 说明 TCP 的流控制算法的示例。图 2A 说明了插入分组的字节流的一部分和每个分组的序列号。该字节流部分的序列号开始于 0。图 2B 说明了接收机 102 怎样建立字节流的副本。虚线方框表示接收窗口，其为了说明目的而被假定具有 1600 的恒定值。

[0035] 在 A 处，分组 1 到达接收机 102，接收机 102 通过向发送机 112 发送具有确认号 (ACK) 1000 和窗口大小 (RevWindow) 1600 的确认分组来确认它。因为 ACK+RevWindow = 2600，所以发送机 112 能够向接收机 102 发送分组 2、3 和 4。然后，发送机 112 向接收机 102 发送分组 2 和 3。

[0036] 在 B 处，分组 3 到达接收机 102，而分组 2 已经在网络链路上被延迟。接收机 102 向发送机 112 发送具有 ACK = 1000 和 RevWindow = 1600 的另一个确认分组。发送机 112 在此时向接收机 102 发送分组 4。

[0037] 在 C 处，分组 4 到达接收机 102，而分组 2 仍然停留 (outstand)。 接收机 102 再次向发送机 112 发送 ACK = 1000 和 RevWindow = 1600 的确认分组。然而，发送机 112 在此时不能向接收机 102 发送分组 5，因为它将把序列号带到大于 ACK+RcvWindow 的 2800。从而，直到接收机 102 向发送机 112 发送新的确认分组为止，发送机 112 都不能向接收机 102 发送别的分组。

[0038] 在 D 处，被延迟的分组 2 现在到达接收机 102。接收机 102 向发送机 112 发送 ACK = 2400 和 RcvWindow = 1600 的确认分组。换言之，接收机 102 滑动或前进接收窗口，以便允许发送机 112 发送更多的数据。这个窗口前进允许发送机 112 发送直到序列号 4000 的数据 (即，ACK+RcvWindow = 4000)。因此，发送机 112 能够向接收机 102 发送分组 5、6 和 7。

[0039] 可以看出，TCP 的流控制算法防止发送机 112 在接收机 102 还没有指出它准备好接收这些附加数据的时候向接收机 102 发送数据。随着接收机 102 和发送机 112 之间的距离和延迟的增加，发送分组与从接收机 102 接收确认之间的 RTT 也增加。接收机 102 和发送机 112 之间的长 RTT 可能导致发送机 112 延迟发送附加分组，以等待下一个来自于接收机 102 的确认。

[0040] 从而在本发明的一个实施例中，控制模块 104 和控制模块 110 进行合作，以同步管理接收机 102 和发送机 112 之间的分组流。根据本发明的实施例，控制模块 104 和 110 进行合作，以增加被指示的窗口并且保持管线被增加到大于或等于网络带宽与接收机 102 和发送机 112 之间的往返延迟之积。从而，本发明的实施例通过为发送机 112 向接收机 102 发送数据提供一个更大的指示窗口来优化 TCP 的流控制。

[0041] 图 3 是根据本发明实施例、由控制模块 104 和 110 来优化的 TCP 的流控制的操作

流程图。与发送机 112 相关联的控制模块 110 双向监视 302 在接收机 102 和发送机 112 之间的一个或多个分组流。特别是，控制模块 110 检查由发送机 112 发送的数据分组的序列号、确认号，以及由接收机 102 发送的确认分组的指示窗口。发送机 112 向接收机 102 发送数据分组。在该数据分组到达接收机 102 之前，控制模块 110 拦截 (intercept)302 数据分组。基于先前的分组流检查，控制模块 110 确定具有序列号的数据分组位于或接近由接收机 102 指示的窗口顶端。 控制模块 110 认为这种数据分组将被″窗口限制″。当控制模块 110 检测窗口限制分组时，它把与这个分组流相关联的当前指示窗口存储在存储区中。其后，控制模块 110 能够用所存储的窗口值来确定由发送机 112 发送的后续数据分组是否被窗口限制。

[0042] 根据本发明的优选实施例，控制模块 110 还把数据分组表征为窗口限制的（例如，通过使用在 TCP 报头中可用的一个或多个字段或者向数据分组添加数据）。控制模块 110 还可以通过使用该窗口指示的最低比特位来表征数据分组，以指示当前的数据分组是否被窗口限制。在控制模块 110 表征窗口限制的数据分组之后，它把被表征的数据分组发送到接收机 102。

[0043] 在被表征的数据分组到达接收机 102 之前，与接收机 102 相关联的控制模块 104 拦截被表征的数据分组，并且通过查看这个分组流被窗口限制的特征而获知。然后，控制模块 104 从数据分组中除去表征，并且把数据分组发送到接收机 102。对于窗口限制的分组流，如果控制模块 104 具有充足的缓存资源，则控制模块 104 还增加由接收机 102 指示的窗口。特别是，当接收机 102 向发送机 112 发送一个或多个确认分组时，控制模块 104 拦截这些确认分组，并且增加 306，在这些确认分组中指示的指示窗口。例如，窗口指示的改变值可能是 64 千字节。根据本发明的示例性实施例，通过在连接建立期间把窗口比例协调到足够高的值，控制模块 104 能够将窗口指示值设置成高达 1,073,725,440 字节。控制模块 104 还设置一个标记，使得对于来自于接收机的后续确认分组，它也增加窗口指示。

[0044] 控制模块 104 向发送机 112 发送调整后的确认分组。发送机 112 接收调整后的确认分组，并且查看确认分组中指示的被调整窗口。由于较大的指示窗口，发送机 112 向接收机 102 发送附加的数据分组，直到被调整窗口所允许的量。在这些数据分组到达接收机 102 之前，控制模块 104 在这些数据分组到达控制模块 104 的时候拦截 308 它们。然后，控制模块 104 确定 310 接收机 102 是否已经将其窗口前进了足够远，使得从发送机 112 到达的特殊数据分组处于接收机 102 当前窗口的范围之内。在本发明的一个实施例中，控制模块 104 能够通过监视从接收机 102 到发送机 112 的分组流来做出这个确定。特别是，控 制模块 104 拦截由接收机 102 发送到发送机 112 的一个或多个分组，并且确定接收机 102 是否已经将其窗口前进足够远，以便容纳来自发送机 112 的数据分组（例如，通过检查确认号和窗口大小）。如果接收机 102 已经将窗口前进足够远以便容纳数据分组，则控制模块 104 直接向接收机 102 发送 312 数据分组，而不将数据分组存储在其缓存中。

[0045] 另一方面，如果接收机 102 还未将窗口前进足够远以容纳数据分组，则控制模块 104 把数据分组存储 314 在其缓存中。当数据分组被存储在缓存中时，控制模块 104 继续监视来自于接收机 102 的确认分组，以便确定接收机 102 后来是否前进了窗口。如果控制模块 104 确定接收机 102 随后已经将窗口前进了足够远，则它把被存储的数据分组发送 316 到接收机 102。在本发明的一个实施例中，如果在数据分组中使用了 TCP 时间戳，则在向接

收机 102 发送数据分组之前，控制模块 104 把数据分组的时间戳调整到如最接近从发送机 112 到接收机 102 的最新时间 (recent)。[0045] 在 TCP 会话的过程中，分组流可能不再被窗口限制。例如，在接收机 102 前进其窗口之后，它向发送机 112 发送一个或多个确认分组。基于确认分组中指示的确认号和窗口大小，控制模块 110 确定从发送机 112 到接收机 102 的数据分组不再具有位于或接近指示窗口顶端的序列号，并由此不被窗口限制。因此，控制模块 110 把从发送机 112 到接收机 102 的一个或多个数据分组表征为不被窗口限制。控制模块 104 在被表征的数据分组到接收机 102 的路上拦截它们，并且察出分组流不再被窗口限制。因此，对于接收机 102 随后发送到发送机 112 的确认分组，控制模块 104 拦截确认分组，通过减小指示窗口来调整确认分组，然后向发送机 112 转发确认分组。在本发明的一个实施例中，每个减少的数量是这样的，即，使得指示窗口的顶端保持不变。控制模块 104 继续减小指示窗口，直到它达到接收窗口的大小为止。而且，控制模块 104 可能继续减小指示窗口，直到分组流变得窗口限制为止，在这种情况下，控制模块 104 然后 110 开始增加指示窗口。

[0046] 根据本发明的实施例，每个时间控制模块 104 或 110 改变 TCP 报头的任意字段，它还调整 TCP 报头的校验和，使得如果校验和在改变之前是正确的则它就是正确的，如果校验和在改变之前是不正确的 则它就是不正确的。

[0047] 此外，本发明的实施例还允许报告控制模块的性能。特别是，与当前的序列号和确认号一起存储在控制模块 110 中的窗口指示被用来估计窗口指示中的增加正在提高数据传输速率（从而不用通过控制模块 104 和 110 的优化就计算数据传输率的估计）的程度。这些估计可用于连续地估算优化功效。然而，这个优化的窗口指示越大，则每秒经过网络的分组更多，而且很难精确地估计相同的分组在没有优化的情况下经过网络的时间。因此，本发明的实施例找出在未使用优化的时候每个分组不得不更小的程度，以便使分组符合原始的窗口大小，并且让分组到达时的速率与经过优化的分组到达的速率相同。从而，如果分组与经过优化之后的以相同的速率到达，则本发明的实施例估计每个分组将更小的程度。因此对于每个分组来说计算如下：

[0048] 未优化的分组大小＝分组的真实大小 *（原始窗口大小 /（分组中最高位的序列号 - 当前的确认号））

[0049] 然而，未优化分组的大小不能够大于分组的当前真实大小。因此，如果上述的计算产生大于分组的当前的真实大小的值，则未优化分组大小的值被设置成分组的当前的真实大小。

[0050] 这个计算得出的数随时间被累积，以估计没有优化的数据传输率。特别是，在（例如 1 秒）周期中未优化的分组的总大小提供了未经优化的数据传输率。并且，该周期中真实分组大小的合计 (total) 提供了经过优化的数据传输率。

[0051] 图 4 是由本发明的一个实施例实现、并由与发送机 112 相关联的控制模块 110 执行的 TCP 优化程序的流程图。控制模块 110 监视 402 发送机 112 和接收机 102 之间的一个或多个分组流。特别是，控制模块 110 检查由发送机 112 发送的数据分组的序列号、以及由接收机 102 发送的确认分组的确认号和窗口指示。控制模块 110 接收 404 来自于发送机 112 的数据分组。然后，控制模块 110 确定 406 从发送机 112 接收机 102 的数据分组是否被窗口限制。例如，如果数据分组携带具有位于或接近接收机 102 指示的窗口顶端的序列

号，则它被窗口限制。如果数据分组不被窗口限制，则控制模块 110 在 402 继续监视发送机 112 和接收机 102 之间的分组流。如果数据分组被窗口限制，则控制模 块 110 把窗口指示存储 408 在存储区中，用作日后参考。

[0052] 控制模块 110 还把数据分组表征 410 为窗口限制，并把被表征的数据分组发送到接收机 102。例如，控制模块 110 可以通过使用 TCP 报头中可用的一个或多个字段、向数据分组添加数据、或使用窗口指示的最低位来表征数据分组。控制模块 110 监视 412 发送机 112 和接收机 102 之间的后续分组流。然后，控制模块 110 确定 414 分组流是否仍然被窗口限制。倘若如此，控制模块 110 在 412 继续监视分组流。否则，控制模块 110 把从发送机 112 到接收机 102 的数据分组表征 416 为没有窗口限制。然后，流程回到 402。

[0053] 图 5 是说明由本发明的一个实施例实现、并由与接收机 102 相关联的控制模块 104 执行的 TCP 优化程序的流程图。控制模块 104 接收 502 来自于发送机 112 的被表征的数据分组。然后，控制模块 104 确定 504 这个表征是否指示数据分组被窗口限制还是不被窗口限制。如果该表征指示不被窗口限制，则控制模块 104 减小 506 从接收机 102 发送到发送机 112 的确认分组中的窗口指示。根据本发明的实施例，窗口指示的最小尺寸是接收窗口的大小。在控制模块 104 减小窗口指示之后，流程回到 502。如果该表征指示被窗口限制，则控制模块 104 确定 508 它是否具有充足的缓存资源。如果控制模块 104 没有充足的缓存资源，则它不改变窗口指示，流程然后回到 502。如果控制模块 104 具有充足的缓存资源，则它增加 510 由接收机 102 发送到发送机 112 的确认分组中的窗口指示。控制模块 104 还设置一个标记，使得对于来自于接收机的后续确认分组而言，它也增加窗口指示。

[0054] 控制模块 104 还确定 512 接收机 102 是否已经将窗口前进了足够远、以便容纳来自于发送机 112 的数据分组。如果接收机 102 已经将窗口前进了足够远，则控制模块 104 直接向接收机 102 发送 514 没有表征的数据分组。然后，流程回到 502。如果接收机 102 还没有将窗口前进足够远，则控制模块把数据分组存储 516 在其缓存中。当窗口被前进足够远时，如由接收机 102 发送的一个或多个确认分组中所指示的，则控制模块 104 向接收机 102 发送 518 没有表征的数据分组。然后，流程回到 502。在本发明的实施例中，如果在数据分组中使用了 TCP 时间戳，则在向接收机 102 发送数据分组之前，控制模块 104 把 数据分组的时间戳调整到最接近从发送机 112 到接收机 102 的最新时间。

[0055] 图 6 示出了被窗口限制并且通过控制模块 104 和 110 进行优化的示例 TCP 事务 (transaction) 的时序图。在图 6 中，数据传输是从发送机 112 到接收机 102。从接收机 102 到发送机 112 的分组是确认分组。此外，一个确认分组被假定为被产生用于每隔一个的 (every second) 数据分组，并且确认分组中指示的窗口大小被假定为四个相等的分组。从图 6 可以看出，当窗口被限制并且发送机 112 和接收机 102 之间的延迟较大时，由发送机 112 发送的数据分组的连续组之间经过了相当长的时间，其原因是由于接收机 102 发射确认分组中的延迟。基于延迟的 TCP 空闲时间导致单个 TCP 流与网络链路上实际可用的情况相比较来说获得较低的吞吐量。因为 TCP 流不能利用全部的现有网络带宽，所以这些未利用的容量转化为较高的传输时间。

[0056] 图 7 示出了被窗口限制但是未通过控制模块 104 和 110 优化的示例 TCP 事务的时序图。在图 7 中，数据传输是从发送机 112 到接收机 102。从接收机 102 到发送机 112 的分组是确认分组。此外，一个确认分组被假定为产生用于每隔一个的数据分组，并且确认分

组中指示的窗口大小被假定为四个相等的分组，但是基于从控制模块 110 接收的逐一分组 (packet-by-packet) 信号由控制模块 104 增加到 16 个分组。在图 7 中可以看出，由于较大的指示窗口尺寸的缘故，发送机 112 能够把大量数据发射到网络中。因此如从图 6 可以区别的那样，数据分组和确认分组的连续组的发送之间没有大量的时间延迟。从而，根据本发明实施例的优化获得了与未使用优化相比较的较高的吞吐量和较短的传输时间。

[0057] 虽然本发明的特殊的实施例和应用已经在此被说明和描述，但是应当理解，本发明不局限于此处所公开的精确的构造和部件，并且在不背离所附的权利要求中定义的本发明的精神和范围的前提下，可以在本发明的方法和设备的布置、操作和细节中进行多种更改、改变和变动。

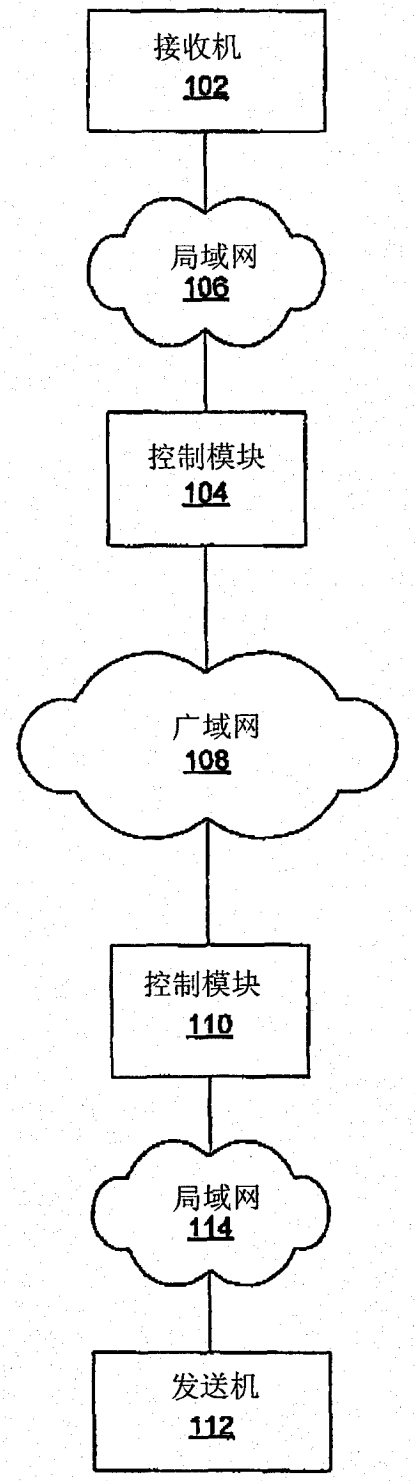


图 1

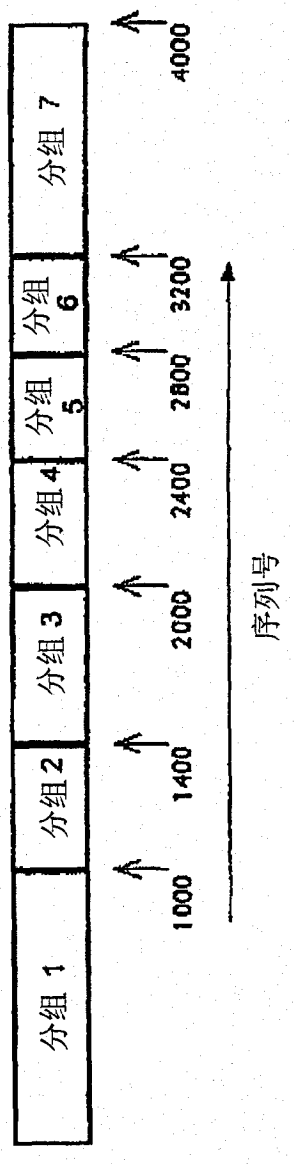

分组 1
分组 2
分组 3
分组 4
分组 5
分组 6
分组 7
1000
1400
2000
2400
2800
3200
4000
序列号

图2A

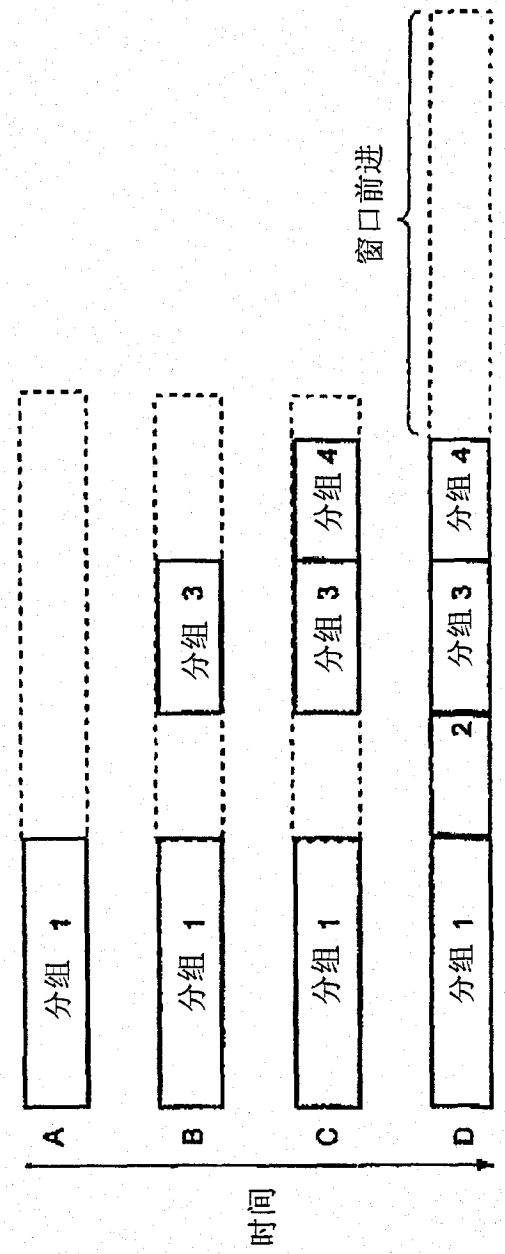

分组 1
分组 1
分组 3
分组 1
分组 3
分组 4
分组 1
分组 2
分组 3
分组 4
A
B
C
D
时间
窗口前进

图2B

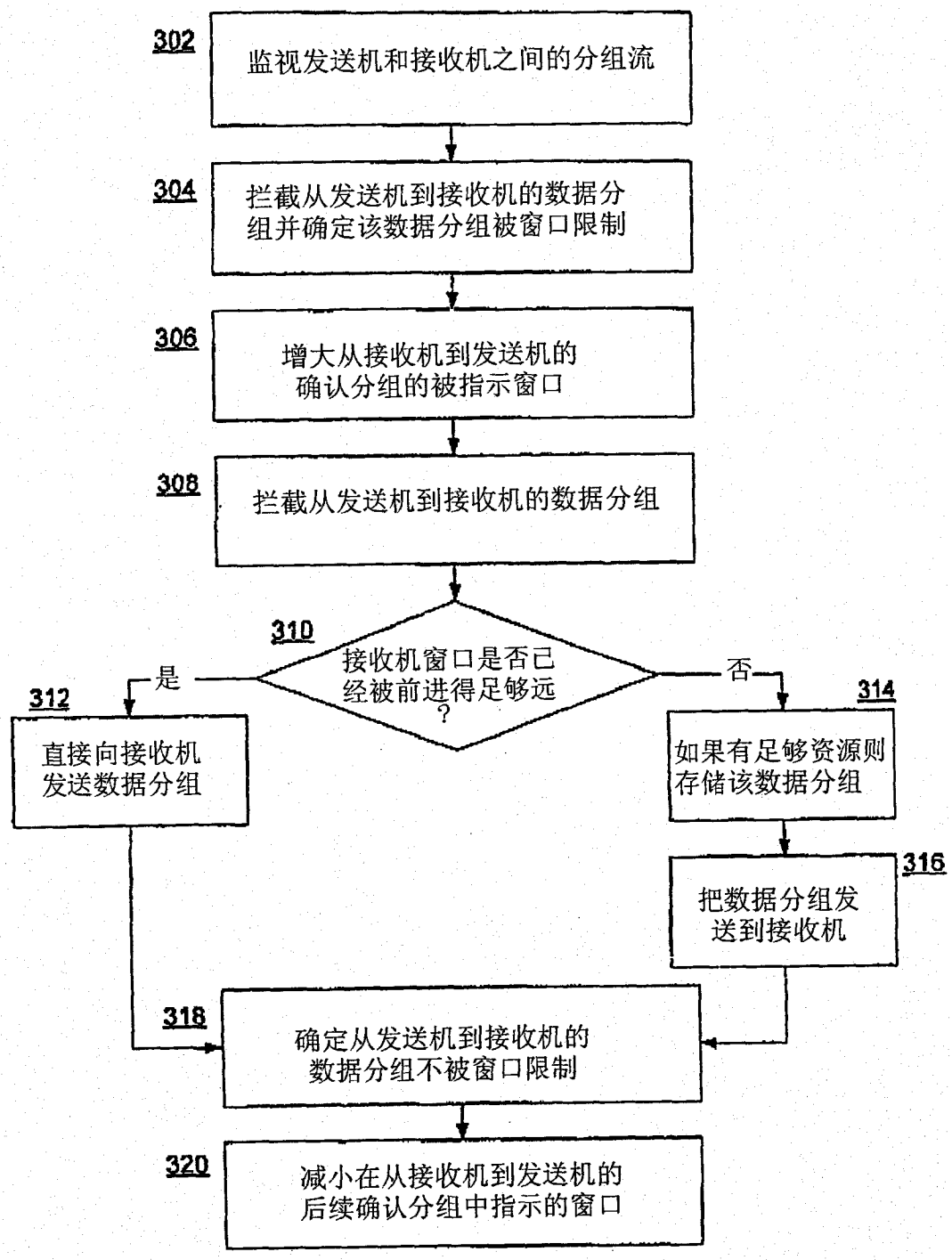

302
监视发送机和接收机之间的分组流
304
拦截从发送机到接收机的数据分组并确定该数据分组被窗口限制
306
增大从接收机到发送机的确认分组的被指示窗口
308
拦截从发送机到接收机的数据分组
310
接收机窗口是否已经被前进得足够远？
是
否
312
直接向接收机发送数据分组
314
如果有足够资源则存储该数据分组
316
把数据分组发送到接收机
318
确定从发送机到接收机的数据分组不被窗口限制
320
减小在从接收机到发送机的后续确认分组中指示的窗口

图 3

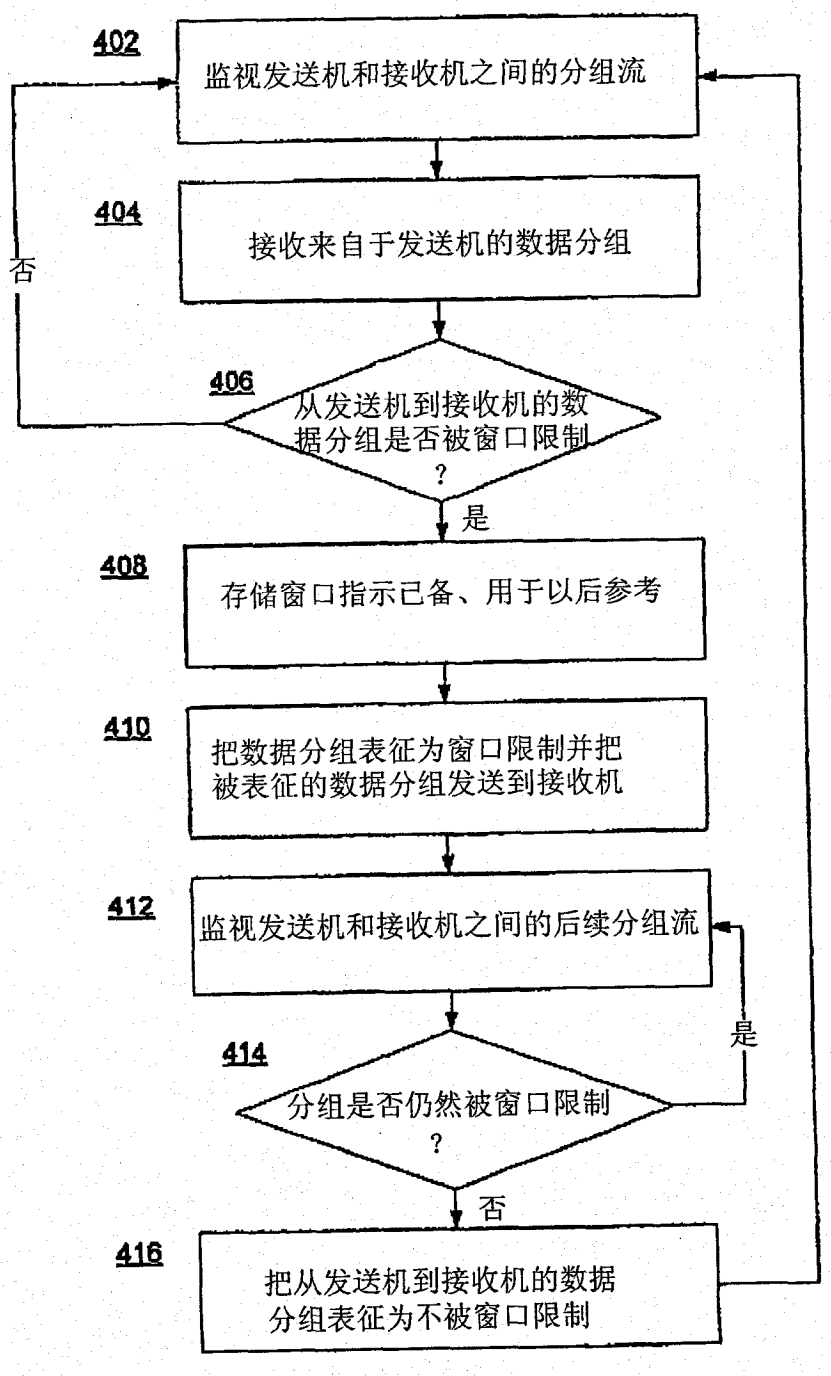

402
监视发送机和接收机之间的分组流
404
接收来自于发送机的数据分组
否
406
从发送机到接收机的数据分组是否被窗口限制？
是
408
存储窗口指示已备、用于以后参考
410
把数据分组表征为窗口限制并把被表征的数据分组发送到接收机
412
监视发送机和接收机之间的后续分组流
414
分组是否仍然被窗口限制？
是
否
416
把从发送机到接收机的数据分组表征为不被窗口限制

图 4

图5

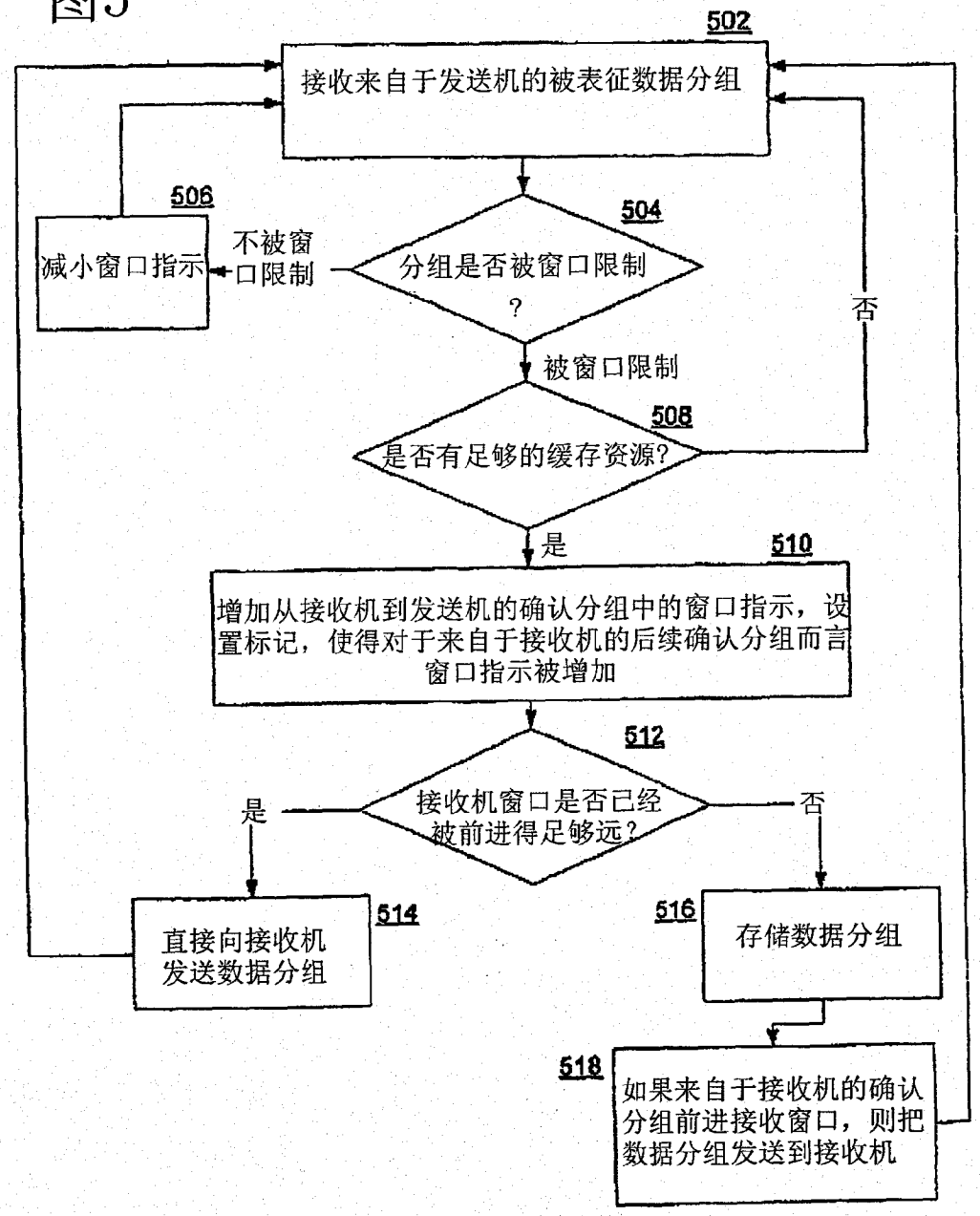
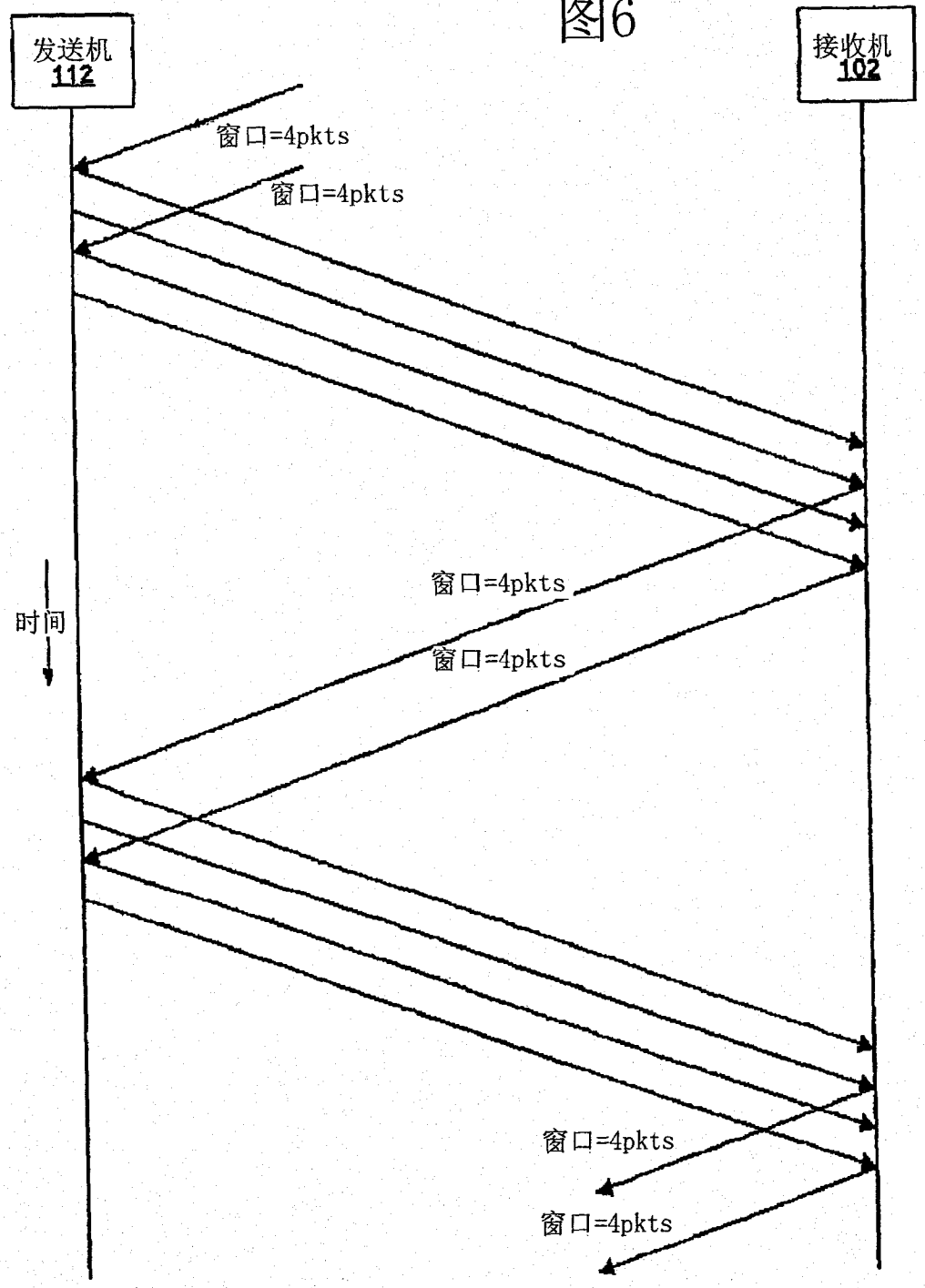

502
接收来自于发送机的被表征数据分组
504
分组是否被窗口限制？
不被窗口限制
506
减小窗口指示
被窗口限制
508
是否有足够的缓存资源?
否
是
510
增加从接收机到发送机的确认分组中的窗口指示，设置标记，使得对于来自于接收机的后续确认分组而言窗口指示被增加
512
接收机窗口是否已经被前进得足够远？
是
否
514
直接向接收机发送数据分组
516
存储数据分组
518
如果来自于接收机的确认分组前进接收窗口，则把数据分组发送到接收机

图6


发送机
112
接收机
102
窗口=4pkts
窗口=4pkts
窗口=4pkts
窗口=4pkts
时间
窗口=4pkts
窗口=4pkts

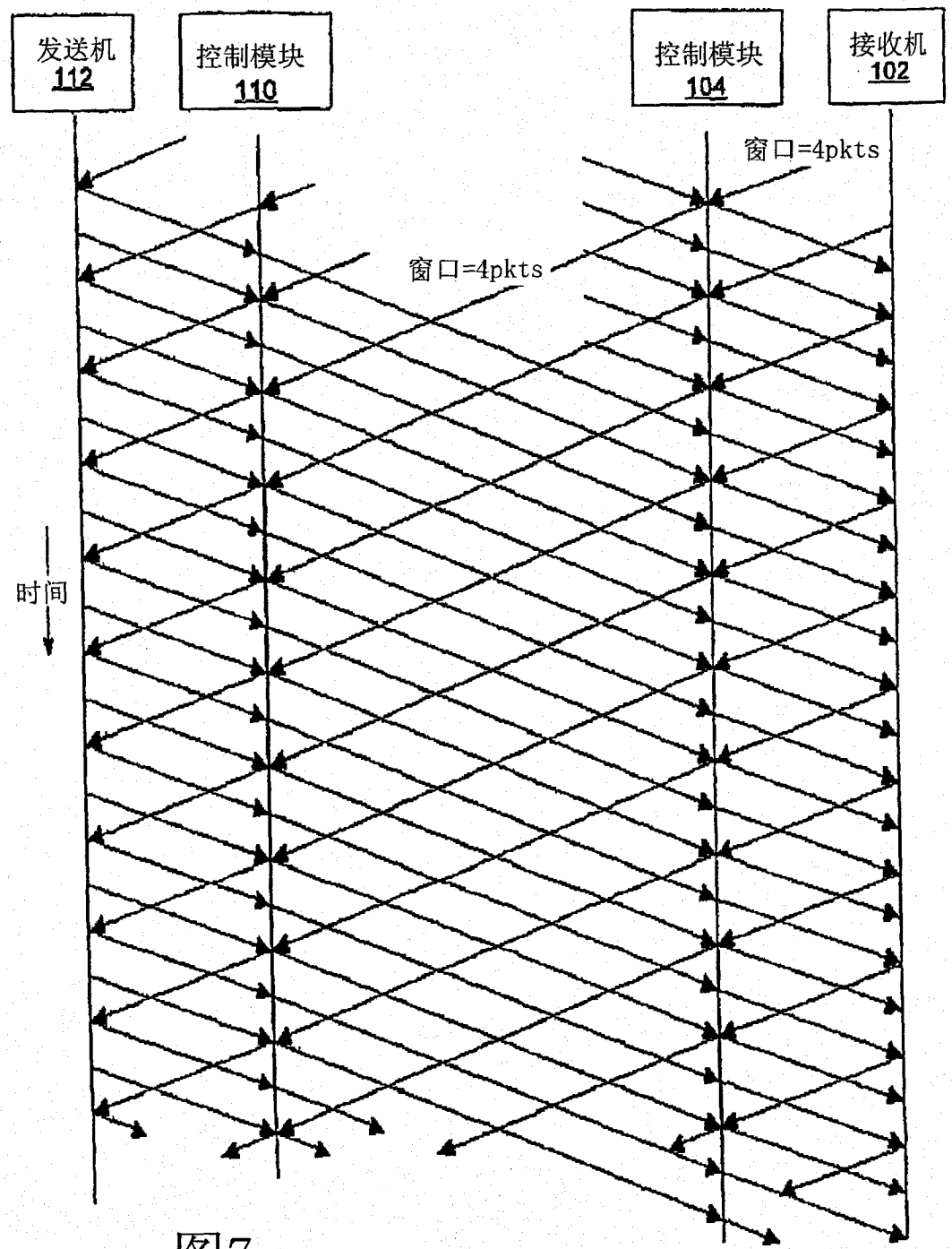


图7